Scatter-Gather Live Migration of Virtual Machines

Kartik Gopalan
Associate Professor, Computer Science

Joint work with
Umesh Deshpande, Yang You, Danny Chan (Students)
Nilton Bila (IBM Research, Yorktown Heights)
Motivation: VM Eviction Time

- Live VM Migration: Transfer of a running VM between hosts
  - VCPU, Memory, (Disk), and (Network state)
- Traditional metrics
  - Downtime, Total Migration Time, Network Overhead, Application performance
- We consider a new metric
  - VM Eviction time: Time to completely evict a VM’s state from the source.
- Why is it important?
  - Quickly eliminate hotspots by moving out VMS
  - Opportunistic power saving by turning off servers
  - Quickly de-provision less important VMs to accommodate more important ones
  - Emergency maintenance; handling imminent failures
Coupling of Source and Destination

- Traditionally
  - Eviction time = Total Migration Time
  - Source cannot eliminate the outgoing VM’s state until target receives the entire VM.
Pre-copy and Post-copy

- **Pre-Copy**
  - 1st Iteration: Transfers entire memory
  - Preparation (live)
  - Downtime

- **Post-Copy**
  - Preparation (live)
  - Downtime
  - Resume Time (live)
  - Active Push + Demand Paging
  - (Non-pageable Memory)
Impact of Destination Resource Pressure

- Source migrating a 5GB idle VM
- Destination running two 5GB VMs running Tunkrank with 4GB memory footprint.
- Eviction time is 6X longer when destination memory = 12GB
- Similar effects when other resources are constrained at the destination
  - CPU cores busy
  - Network interface under contention
State of the Art

- Lowering Total Migration Time
  - Ballooning, compression, dropping the guest cache, deduplication
  - Orthogonal to our approach

- Checkpoint/Restore
  - Typically non-live; restore follows checkpoint; large downtime
    - Remus does high-frequency checkpointing for high availability; quick restoration but large runtime overhead for write-intensive apps; doubles memory usage.

- Post-copy Migration
  - Quickly offloads VM’s CPU state to destination.
    - Memory follows from the source
  - Snowflock, Jettison, Reactive consolidation use post-copy
Solution: Scatter-Gather VM Migration

- Scatter VM memory to intermediaries, Gather VM memory from intermediaries
- Intermediaries could be hosts at the destination rack, memory appliances, middleboxes, etc.
- Scatter-Gather = Post-copy variant + live checkpoint/restore via intermediate nodes
  - Concurrent Scatter (checkpoint) and Gather (restore) phases
  - Post-copy variant
    - Pre-paging via intermediaries
    - Page-faults serviced from source/intermediaries
Implementation

- Source
  - Scatters pages to VMD;
    - Sends IDs of scattered pages to destination
  - Services faults from destination
  - Prioritizes fault servicing over scatter
- Destination
  - Gathers pages from VMD
  - Requests faulted pages
    - from source while source is scattering, from VMD afterwards
- Virtualized Memory Device (VMD) Layer
  - Peer-to-peer memory sharing system over Ethernet
  - Presents the aggregated free memory of intermediaries as a block device to Migration Managers
Preliminary Results

- **Goal:**
  - When destination is resource constrained, Scatter-Gather Migration can deliver lower eviction time than pre-copy or post-copy

- **Setup**
  - Dual quad-core servers with 1.7 GHz CPUs and 16GB DRAM
  - 1Gbps links to a Nortel 4526-GTX switch
  - Host runs Linux 2.6.32 KVM/QEMU 1.6.50; VMs run Linux 3.2.
  - Standard pre-copy implementation in QEMU
  - Post-copy implementation from Yabusame project.
    - Modified for Scatter-Gather
Scatter-Gather delivers constant eviction time with increasing memory pressure.

TMT higher by about 10% since VMD transmission protocol delivers lower throughput (750 Mbps) than TCP in our current implementation.
Bandwidth Pressure at the Destination: Tradeoff between Eviction Time and Application Performance

Memcached request latency when migrating a 5GB VM

Two 5GB VMs at destination servicing memcached requests from external client

<table>
<thead>
<tr>
<th>Rate Limit (256 Mbps)</th>
<th>Eviction Time (Seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pre-copy</td>
<td>160.8</td>
</tr>
<tr>
<td>Post-copy</td>
<td>98.6</td>
</tr>
<tr>
<td>Scatter-Gather (256 Mbps)</td>
<td>98.6</td>
</tr>
</tbody>
</table>

TABLE I
Eviction time comparison when the migrating 5GB idle VM with and without rate limiting.
Conclusions

- Scatter-Gather Live VM migration
  - Reduces eviction time without affecting total migration time
  - Network overhead can be tackled using compression/deduplication
  - Bigger impact when migrating multiple VMs together
  - Leads to the idea of permanently “scattered” VMs
  - Removes memory as a bottleneck for consolidation
  - Provides greater agility in scaling out when demand increases.
Thanks!

Contact:
Kartik Gopalan
kartik@binghamton.edu